

**Government Smart Card  
Interoperability Specification v2.1  
(NISTIR 6887 - 2003 Edition)  
Basic Services Interface  
Java Binding**

**Conformance Test Assertions**

**FINAL DRAFT**

**Alan Goldfine  
October 8, 2004**

This document contains the conformance test assertions for the methods comprising the Java language binding of the Basic Services Interface of version 2.1 of the Government Smart Card Interoperability Specification (GSC-IS), as contained in NIST Interagency Report 6887 - 2003 Edition.

The 23 sections of this document correspond to the 23 Java methods specified in Appendix F of the GSC-IS, which in turn correspond to the 23 BSI functions specified in Chapter 4 in the BSI. The test assertions for each of the methods, except for those in section F.2.9 (q.v.), are numbered in the form X.Y, where X is the method number (corresponding to the function order in section 4.1), and Y is the number of the assertion for that function. Thus, 7.4 is the number given to assertion 4 for method 7 (`gscBsiUtilGetCardProperties()`).

Variable names have the form Zxy. Here, Z is the name of the variable, and x and y are the two-digit representations of X and Y, where X.Y is the test assertion within which the variable is defined. x and y have leading zeroes, if necessary. For example, `strctAuthenticator0104` is one of the variables defined in assertion 4 for method 1. Variables that are defined in the "Starting state for each Assertion", and are therefore global across all the assertions for the given function, have y = 00.

Conformance test assertions are statements of behavior, action, or condition that can be measured or tested. The assertions here are derived from the GSC-IS, and bridge the gap between the narrative of the GSC-IS and the test cases that comprise the BSI Java language binding conformance test suite. Each test assertion is an independent, complete, and testable statement, and may result in one or more test cases. The test cases themselves are described in "Government Smart Card Interoperability Specification v2.1 (NISTIR 6887 - 2003 Edition), Basic Services Interface, Java Binding: Conformance Test Instantiation, Verification, and Reporting Scenarios."

## 1. gscBsiUtilAcquireContext()

```
public abstract void gscBsiUtilAcquireContext(  
    int          hCard,  
    String       AID,  
    Vector       strctAuthenticator  
) throws gov.gsc.classes.BSIException;
```

strctAuthenticator is a Vector of BSIAuthenticator objects.  
The BSIAuthenticator class contains the fields

|                  |                  |
|------------------|------------------|
| protected int    | accessMethodType |
| protected int    | keyIDOrReference |
| protected byte[] | AuthValue        |

### References:

1. GSC-IS 4.5.3.
2. GSC-IS F.2.1.

### Starting state for each Assertion:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0100.
2. The card has a target container.
3. There exists a Vector strctAuthenticator0100, containing the authenticators associated with the target container.
4. There is no authenticated session established with any container on the connected card.

### **Assertion 1.1**

Purpose: To test gscBsiUtilAcquireContext() using valid parameters.

### Scenario:

1. A gscBsiUtilAcquireContext() call is made to the SPS, with
  - hCard == hCard0100
  - AID == the AID value of the target container
  - strctAuthenticator == strctAuthenticator0100.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or BSI\_TERMINAL\_AUTH (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_TERMINAL\_AUTH).
2. An authenticated session is established with the target container.

### **Assertion 1.2**

Purpose: To test gscBsiUtilAcquireContext() using a bad handle.

### Scenario:

1. A gscBsiUtilAcquireContext() call is made to the SPS, with

- hCard /= hCard0100
- AID == the AID value of the target container
- strctAuthenticator == strctAuthenticator0100.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE).
2. An authenticated session is not established with the target container.

**Assertion 9.1.2.1**

**Assertion 9.1.2.2**

(See section 9.)

**Assertion 1.3**

Purpose: To test gscBsiUtilAcquireContext() using a bad AID value.

Scenario:

1. A gscBsiUtilAcquireContext() call is made to the SPS, with
  - hCard == hCard0100
  - AID == a String that does not contain the correct AID for any container on the connected card
  - strctAuthenticator == strctAuthenticator0100.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_AID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID).

**Assertion 9.1.3.1**

**Assertion 9.1.3.2**

(See section 9.)

**Assertion 1.4**

Purpose: To test gscBsiUtilAcquireContext() using an authentication method that is not available on the card.

Scenario:

1. A Vector strctAuthenticator0104 of authenticators, at least one of which is not associated with the target container, is declared.
2. A gscBsiUtilAcquireContext() call is made to the SPS, with
  - hCard == hCard0100
  - AID == the AID value of the target container
  - strctAuthenticator == strctAuthenticator0104.

Expected Results:

1. The call returns
  - the return code BSI\_ACR\_NOT\_AVAILABLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_ACR\_NOT\_AVAILABLE).
2. An authenticated session is not established with the target container.

**Assertion 9.1.4.1**

**Assertion 9.1.4.2**

(See section 9.)

**Assertion 1.5**

Purpose: To test gscBsiUtilAcquireContext() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A gscBsiUtilAcquireContext() call is made to the SPS, with
  - hCard == hCard0100
  - AID == the AID value of the target container
  - strctAuthenticator == strctAuthenticator0100.

Expected Results:

1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED).

**Assertion 9.1.5.1**

**Assertion 9.1.5.2**

(See section 9.)

**Assertion 1.6**

Purpose: To test gscBsiUtilAcquireContext() using a bad authenticator.

Scenario:

1. A vector strctAuthenticator0106 of authenticators, at least one of which is invalid, is declared.
2. A gscBsiUtilAcquireContext() call is made to the SPS, with
  - hCard == hCard0100
  - AID == the AID value of the target container
  - strctAuthenticator == strctAuthenticator0106.

Expected Results:

1. The call returns

- the return code BSI\_BAD\_AUTH (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AUTH).
2. An authenticated session is not established with the target container.

#### **Assertion 9.1.6.1**

#### **Assertion 9.1.6.2**

(See section 9.)

#### **Assertion 1.7**

Purpose: To test gscBsiUtilAcquireContext() with a removed card.

Scenario:

1. The connected card is removed from the reader.
2. A gscBsiUtilAcquireContext() call is made to the SPS, with
  - hCard == hCard0100
  - AID == the AID value of the target container
  - strctAuthenticator == strctAuthenticator0100.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_REMOVED).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

#### **Assertion 1.8**

Purpose: To test gscBsiUtilAcquireContext() with a blocked PIN.

Scenario:

1. The PIN for the target container is blocked.
2. A gscBsiUtilAcquireContext() call is made to the SPS, with
  - hCard == hCard0100
  - AID == the AID value of the target container
  - strctAuthenticator == strctAuthenticator0100.

Expected Results:

1. The call returns
  - the return code BSI\_PIN\_BLOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_PIN\_BLOCKED).
2. An authenticated session is not established with the target container.

#### **Assertion 9.1.8.1**

**Assertion 9.1.8.2**

(See section 9.)

## 2. gscBsiUtilConnect()

```
public abstract int gscBsiUtilConnect(  
    String      readerName  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.5.4.
2. GSC-IS F.2.2.

### Starting state for each Assertion:

1. A card that claims conformance to the GSC-IS is in a particular reader available to the candidate SPS.
2. The card is not connected to the reader.
3. The name of the reader is represented by the String readerName0200.
4. There exists an int hCard0200.

### **Assertion 2.1**

Purpose: To test gscBsiUtilConnect() using a good card inserted into a specified reader.

### Scenario:

1. An hCard0200 = gscBsiUtilConnect() call is made to the SPS, with
  - readerName == readerName0200.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown)
  - hCard0200 == a valid handle.
2. The card is connected, with handle hCard0200, to the reader readerName0200.

### **Assertion 2.2**

Purpose: To test gscBsiUtilConnect() using a good card inserted into a non-specified reader.

### Scenario:

1. An hCard0200 = gscBsiUtilConnect() call is made to the SPS, with
  - readerName == "".

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown)
  - hCard0200 == a valid handle.
2. The card is connected, with handle hCard0200, to the first available reader.

### Assertion 2.3

Purpose: To test gscBsiUtilConnect() using a bad reader name.

Scenario:

1. An hCard0200 = gscBsiUtilConnect() call is made to the SPS, with
  - readerName == a String that does not represent a valid reader.

Expected Results:

1. The call returns
  - the return code BSI\_UNKNOWN\_READER (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_UNKNOWN\_READER).

*Note: There are no gscBsiUtilGetExtendedErrorText() assertions following Assertion 2.3 because the card is presumably not connected.*

### Assertion 2.4

Purpose: To test gscBsiUtilConnect() with no card in the reader.

Scenario:

1. An hCard0200 = gscBsiUtilConnect() call is made to the SPS, with
  - readerName == readerName0200.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_ABSENT (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_ABSENT).

*Note: There are no gscBsiUtilGetExtendedErrorText() assertions following Assertion 2.4 because the card is presumably not connected.*

### Assertion 2.5

Purpose: To test gscBsiUtilConnect() using a bad inserted card.

Scenario:

1. A card that does not claim conformance to the GSC-IS is in a particular reader available to the candidate SPS.
2. An hCard0200 = gscBsiUtilConnect() call is made to the SPS, with
  - readerName == readerName0200.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_ABSENT or BSI\_UNKNOWN\_ERROR (a BSIException is thrown, with BSIException.getErrorCode returning either BSI\_CARD\_ABSENT or BSI\_UNKNOWN\_ERROR).

*Note: There are no gscBsiUtilGetExtendedErrorText() assertions following Assertion 2.5 because the card is presumably not connected.*



### 3. gscBsiUtilDisconnect()

```
public abstract void gscBsiUtilDisconnect(  
    int hCard  
) throws gov.gsc.classes.BSIException;
```

#### References:

1. GSC-IS 4.5.5.
2. GSC-IS F.2.3.

#### Starting state for each Assertion:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0300.

#### **Assertion 3.1**

Purpose: To test gscBsiUtilDisconnect() using a good handle.

#### Scenario:

1. A gscBsiUtilDisconnect() call is made to the SPS, with hCard == hCard0300.

#### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown).
2. The card is disconnected.

#### **Assertion 3.2**

Purpose: To test gscBsiUtilDisconnect() using a bad handle.

#### Scenario:

1. A gscBsiUtilDisconnect() call is made to the SPS, with hCard /= hCard0300.

#### Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE).
2. The card is still connected.

#### **Assertion 9.3.2.1**

#### **Assertion 9.3.2.2**

(See section 9.)

#### **Assertion 3.3**

Purpose: To test gscBsiUtilDisconnect() with a removed card.

Scenario:

1. The connected card is removed.
2. A gscBsiUtilDisconnect() call is made to the SPS, with  
hCard == hCard0300.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with  
BSIException.getErrorCode returning BSI\_CARD\_REMOVED).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

#### 4. gscBsiUtilBeginTransaction()

```
public abstract void gscBsiUtilBeginTransaction(  
    int          hCard,  
    boolean      blType  
) throws gov.gsc.classes.BSIException;
```

##### References:

1. GSC-IS 4.5.6.
2. GSC-IS F.2.4.

##### Starting state for each Assertion:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0400.

##### **Assertion 4.1**

Purpose: To test gscBsiUtilBeginTransaction() as a blocking transaction call, with no existing transaction lock, using valid parameters.

##### Scenario:

1. There is no existing transaction lock.
2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
  - hCard == hCard0400
  - blType == true.

##### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_SPSSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_SPSSERVICE).
2. If the return code is BSI\_OK, a transaction is established with the smart card.

##### **Assertion 4.2**

Purpose: To test gscBsiUtilBeginTransaction() as a non-blocking transaction call, with no existing transaction lock, using valid parameters.

##### Scenario:

1. There is no existing transaction lock.
2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
  - hCard == hCard0400
  - blType == false.

##### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_SPSSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_SPSSERVICE).

2. If the return code is BSI\_OK, a transaction is established with the smart card.

#### **Assertion 4.3**

Purpose: To test gscBsiUtilBeginTransaction() as a non-blocking transaction call, with another application having an existing transaction lock, using valid parameters.

Scenario:

1. Another application has established a transaction lock.
2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
  - hCard == hCard0400
  - blType == false.

Expected Results:

1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED).

#### **Assertion 9.4.3.1**

#### **Assertion 9.4.3.2**

(See section 9.)

#### **Assertion 4.4**

Purpose: To test gscBsiUtilBeginTransaction() as a blocking transaction call, with another application having established a transaction lock, using valid parameters.

Scenario:

1. Another application has established a transaction lock.
2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
  - hCard == hCard0400
  - blType == true.

Expected Results:

1. The call waits indefinitely.

#### **Assertion 4.5**

Purpose: To test gscBsiUtilBeginTransaction() as a non-blocking transaction call, with the current application having already established a transaction lock, using valid parameters.

Scenario:

1. The current application has established a transaction lock.
2. A gscBsiUtilBeginTransaction() call is made to the SPS, with

- hCard == hCard0400
- blType == false.

Expected Results:

1. The call returns
  - the return code BSI\_NOT\_TRANSACTED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NOT\_TRANSACTED).

**Assertion 9.4.5.1**

**Assertion 9.4.5.2**

(See section 9.)

**Assertion 4.6**

Purpose: To test gscBsiUtilBeginTransaction() as a blocking transaction call, with the current application having already established a transaction lock, using valid parameters.

Scenario:

1. The current application has established a transaction lock.
2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
  - hCard == hCard0400
  - blType == true.

Expected Results:

1. The call returns
  - the return code BSI\_NOT\_TRANSACTED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NOT\_TRANSACTED).

**Assertion 9.4.6.1**

**Assertion 9.4.6.2**

(See section 9.)

**Assertion 4.7**

Purpose: To test gscBsiUtilBeginTransaction() as a blocking transaction call, with no existing transaction lock, with a bad handle.

Scenario:

1. There is no existing transaction lock.
2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
  - hCard /= hCard0400
  - blType == true.

Expected Results:

1. The call returns

- the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return code BSI\_NO\_SPSSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_SPSSERVICE).

2. No transaction is established with the smart card.

**Assertion 9.4.7.1**

**Assertion 9.4.7.2**

(See section 9.)

## 5. gscBsiUtilEndTransaction()

```
public abstract void gscBsiUtilEndTransaction(  
    int hCard  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.5.7.
2. GSC-IS F.2.5.

### Starting state for each Assertion:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0500.

### **Assertion 5.1**

Purpose: To test gscBsiUtilEndTransaction() with an existing transaction lock, using valid parameters.

### Scenario:

1. The current application has established a transaction lock.
2. A gscBsiUtilEndTransaction() call is made to the SPS, with
  - hCard == hCard0500.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_SPSSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_SPSSERVICE).
2. If the return code is BSI\_OK, the previously existing transaction lock is ended.

### **Assertion 5.2**

Purpose: To test gscBsiUtilEndTransaction() with no existing transaction lock.

### Scenario:

1. There is no existing transaction lock.
2. A gscBsiUtilEndTransaction() call is made to the SPS, with
  - hCard == hCard0500.

### Expected Results:

1. The call returns
  - the return code BSI\_NOT\_TRANSACTED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NOT\_TRANSACTED) or the return code BSI\_NO\_SPSSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_SPSSERVICE).

### **Assertion 9.5.2.1**

#### **Assertion 9.5.2.2**

(See section 9.)

#### **Assertion 5.3**

Purpose: To test gscBsiUtilEndTransaction() with a bad handle.

Scenario:

1. The current application has established a transaction lock.
2. A gscBsiUtilEndTransaction() call is made to the SPS, with
  - hCard /= hCard0500.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return code BSI\_NO\_SPSSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_SPSSERVICE).
2. If the return code is BSI\_BAD\_HANDLE, the previously existing transaction lock remains in effect.

#### **Assertion 9.5.3.1**

#### **Assertion 9.5.3.2**

(See section 9.)



## 6. gscBsiUtilGetVersion()

```
public abstract String gscBsiUtilGetVersion()  
    throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.5.8.
2. GSC-IS F.2.6.

### **Assertion 6.1**

Purpose: To test gscBsiUtilGetVersion() using valid parameters.

### Scenario:

1. A String version0600 is declared.
2. A version0600 = gscBsiUtilGetVersion() call is made to the SPS.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown)
  - version0600 == the BSI implementation version of the SPS.

## 7. gscBsiUtilGetCardProperties()

```
public abstract CardProperties gscBsiUtilGetCardProperties(  
    int          hCard  
) throws gov.gsc.classes.BSIException;
```

The CardProperties class contains the fields

```
protected int      cardCapability  
protected byte[]   CCCUniqueID
```

### References:

1. GSC-IS 4.5.9.
2. GSC-IS F.2.7.

### Starting state for each Assertion:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0700.
2. There exists a CardProperties object cardProps0700, with fields
  - int cardCapability0700
  - byte[] CCCUniqueID0700.

### **Assertion 7.1**

Purpose: To test gscBsiUtilGetCardProperties() using valid parameters.

### Scenario:

1. A cardProps0700 = gscBsiUtilGetCardProperties() call is made to the SPS, with
  - hCard == hCard0700.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. If the return code is BSI\_OK, then
  - cardCapability0700 == one of the recognized bitwise masks identifying the provider of the connected card
  - CCCUniqueID0700 == the Card Capability Container ID.

### **Assertion 7.2**

Purpose: To test gscBsiUtilGetCardProperties() using a bad handle.

### Scenario:

1. A cardProps0700 = gscBsiUtilGetCardProperties() call is made to the SPS, with
  - hCard != hCard0700.

### Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return

code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

#### **Assertion 9.7.2.1**

#### **Assertion 9.7.2.2**

(See section 9.)

### **Assertion 7.3**

Purpose: To test gscBsiUtilGetCardProperties() with a removed card.

Scenario:

1. The connected card is removed from the reader.
2. A cardProps0700 = gscBsiUtilGetCardProperties() call is made to the SPS, with
  - hCard == hCard0700.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_REMOVED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

### **Assertion 7.4**

Purpose: To test gscBsiUtilGetCardProperties() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A cardProps0700 = gscBsiUtilGetCardProperties() call is made to the SPS, with
  - hCard == hCard0700.

Expected Results:

1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

#### **Assertion 9.7.4.1**

#### **Assertion 9.7.4.2**

(See section 9.)

## 8. gscBsiUtilGetCardStatus()

```
public abstract void gscBsiUtilGetCardStatus(  
    int hCard  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.5.10.
2. GSC-IS F.2.8.

### Starting state for each Assertion:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0800.

### **Assertion 8.1**

Purpose: To test gscBsiUtilGetCardStatus() using valid parameters.

### Scenario:

1. A gscBsiUtilGetCardStatus() call is made to the SPS, with
  - hCard == hCard0800.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown).

### **Assertion 8.2**

Purpose: To test gscBsiUtilGetCardStatus() using a bad handle.

### Scenario:

1. A gscBsiUtilGetCardStatus() call is made to the SPS, with
  - hCard /= hCard0800.

### Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE).

### **Assertion 9.8.2.1**

### **Assertion 9.8.2.2**

(See section 9.)

### **Assertion 8.3**

Purpose: To test gscBsiUtilGetCardStatus() with a removed card.

### Scenario:

1. The card is removed.
2. A gscBsiUtilGetCardStatus() call is made to the SPS, with

- `hCard == hCard0800.`

Expected Results:

1. The call returns

- the return code `BSI_CARD_REMOVED` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_CARD_REMOVED`).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of `GetExtendedErrorText()`.*

## 9. gscBsiUtilGetExtendedErrorText()

```
public abstract String gscBsiUtilGetExtendedErrorText(  
    int hCard  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.5.11.
2. GSC-IS F.2.9.

### Starting state for each Assertion:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0900.
2. A String errorText0900 is declared.

*Note: The function gscBsiUtilGetExtendedErrorText() can be called following many, but not all, of the other test assertion function calls. gscBsiUtilGetExtendedErrorText() requires, for example, that a card be connected to the reader. Since it is also unclear whether gscBsiUtilGetExtendedErrorText() needs to be capable of providing extended messages for the "successful" return codes BSI\_OK or BSI\_TERMINAL\_AUTH, the function will not be tested for successful codes.*

*For those assertions X.Y for which calling gscBsiUtilGetExtendedErrorText() is meaningful, the gscBsiUtilGetExtendedErrorText() assertions are, in this document, specified immediately following the X.Y assertion as 9.X.Y.1 and 9.X.Y.2.*

### **Assertion 9.X.Y.1**

Purpose: To test gscBsiUtilGetExtendedErrorText(), using valid parameters, following a test corresponding to Assertion X.Y.

### Scenario:

1. A test corresponding to Assertion X.Y has just been completed.
2. An errorText0900 = gscBsiUtilGetExtendedErrorText() call is made to the SPS, with
  - hCard == hCard0900.

### Expected Results:

1. The call returns
  - either the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_TEXT\_AVAILABLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_TEXT\_AVAILABLE)
  - if BSI\_OK is the code returned, then ErrorText0900 == an extended error message
  - if BSI\_NO\_TEXT\_AVAILABLE is the code returned, then ErrorText0900== "".

### **Assertion 9.X.Y.2**

Purpose: To test gscBsiUtilGetExtendedErrorText(), using a bad handle, following each test corresponding to Assertion X.Y.

Scenario:

1. A test corresponding to Assertion X.Y has just been completed.
2. An `errorText0900 = gscBsiUtilGetExtendedErrorText()` call is made to the SPS, with
  - `hCard != hCard0900`.

Expected Results:

1. The call returns
  - the return code `BSI_BAD_HANDLE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_TEXT_AVAILABLE`).



## 10. `gscBsiUtilGetReaderList()`

```
public abstract Vector gscBsiUtilGetReaderList()  
    throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.5.12.
2. GSC-IS F.2.10.

### Starting state for each Assertion:

1. A Vector `vReaderList1000` is declared.

### **Assertion 10.1**

Purpose: To test `gscBsiUtilGetReaderList()` using valid parameters.

### Scenario:

1. A `vReaderList1000 = gscBsiUtilGetReaderList()` call is made to the SPS.

### Expected Results:

1. The call returns
  - the return code `BSI_OK` (no `BSIException` is thrown)
  - `vReaderList1000 ==` a Vector of Strings representing the available readers.

## 11. gscBsiUtilPassthru()

```
public abstract byte[] gscBsiUtilPassthru(  
    int          hCard,  
    byte[]       cardCommand  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.5.13.
2. GSC-IS E.2.11.

### Starting state for each Assertion:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1100.
2. An array of bytes cardResponse1100 is declared.

### **Assertion 11.1**

Purpose: To test gscBsiUtilPassthru() using valid parameters.

### Scenario:

1. A cardResponse1100 = gscBsiUtilPassthru() call is made to the SPS, with
  - hCard == hCard1100
  - cardCommand == an array of bytes representing an APDU.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown)
  - cardResponse1100 == a byte[] containing the APDU response from the connected card.

### **Assertion 11.2**

Purpose: To test gscBsiUtilPassthru() using a bad handle.

### Scenario:

1. A cardResponse1100 = gscBsiUtilPassthru() call is made to the SPS, with
  - hCard != hCard1100
  - cardCommand == an array of bytes representing an APDU.

### Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE).

### **Assertion 9.11.2.1**

### **Assertion 9.11.2.2**

(See section 9.)

### **Assertion 11.3**

Purpose: To test gscBsiUtilPassthru() using a bad cardCommand.

Scenario:

1. A cardResponse1100 = gscBsiUtilPassthru() call is made to the SPS, with
  - hCard == hCard1100
  - cardCommand == an array of bytes that does not represent an APDU.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_PARAM (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_PARAM).

**Assertion 9.11.3.1**

**Assertion 9.11.3.2**

(See section 9.)

#### **Assertion 11.4**

Purpose: To test gscBsiUtilPassthru() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A cardResponse1100 = gscBsiUtilPassthru() call is made to the SPS, with
  - hCard == hCard1100
  - cardCommand == an array of bytes representing an APDU.

Expected Results:

1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED).

**Assertion 9.11.4.1**

**Assertion 9.11.4.2**

(See section 9.)

#### **Assertion 11.5**

Purpose: To test gscBsiUtilPassthru() with a removed card.

Scenario:

1. The connected card is removed from the reader.
2. A cardResponse1100 = gscBsiUtilPassthru() call is made to the SPS, with
  - hCard == hCard1100
  - cardCommand == an array of bytes representing an APDU.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_REMOVED).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

## 12. gscBsiUtilReleaseContext()

```
public abstract void gscBsiUtilReleaseContext(  
    int          hCard,  
    String       AID  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.5.14.
2. GSC-IS F.2.12.

### Starting state for each Assertion:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1200.
2. The card has a target container.
3. An authenticated session has been established with the target container.

### **Assertion 12.1**

Purpose: To test gscBsiUtilReleaseContext() using valid parameters.

### Scenario:

1. A gscBsiUtilReleaseContext() call is made to the SPS, with
  - hCard == hCard1200
  - AID == the AID value of the target container.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown).
2. There is no longer an authenticated session established with the target container.

### **Assertion 12.2**

Purpose: To test gscBsiUtilReleaseContext() using a bad handle.

### Scenario:

1. A gscBsiUtilReleaseContext() call is made to the SPS, with
  - hCard != hCard1200
  - AID == the AID value of the target container.

### Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE).
2. There continues to be an authenticated session established with the target container.

### **Assertion 9.12.2.1**

#### **Assertion 9.12.2.2**

(See section 9.)

#### **Assertion 12.3**

Purpose: To test gscBsiUtilReleaseContext() using a bad AID value.

Scenario:

1. A gscBsiUtilReleaseContext() call is made to the SPS, with
  - hCard == hCard1200
  - AID == a String that does not contain the correct AID for any container on the connected card.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_AID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID).
2. There continues to be an authenticated session established with the target container.

#### **Assertion 9.12.3.1**

#### **Assertion 9.12.3.2**

(See section 9.)

#### **Assertion 12.4**

Purpose: To test gscBsiUtilReleaseContext() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A gscBsiUtilReleaseContext() call is made to the SPS, with
  - hCard == hCard1200
  - AID == the AID value of the target container.

Expected Results:

1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED).

#### **Assertion 9.12.4.1**

#### **Assertion 9.12.4.2**

(See section 9.)

## Assertion 12.5

Purpose: To test gscBsiUtilReleaseContext() with a removed card.

Scenario:

1. The connected card is removed from the reader.
2. A gscBsiUtilReleaseContext() call is made to the SPS, with
  - hCard == hCard1200
  - AID == the AID value of the target container.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

### 13. gscBsiGcDataCreate()

```
public abstract void gscBsiGcDataCreate(  
    int          hCard,  
    String       AID,  
    short        tag,  
    byte[]       dValue  
) throws gov.gsc.classes.BSIException;
```

#### References:

1. GSC-IS 4.6.1.
2. GSC-IS F.3.1.

#### **Assertion 13.1**

Purpose: To test gscBsiGcDataCreate() using valid parameters.

#### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1301
  - the card has a target container
  - an authenticated session has been established with the target container.
2. A gscBsiGcDataCreate() call is made to the SPS, with
  - hCard == hCard1301
  - AID == the AID value of the target container
  - tag == a short whose value is not the tag of a data item in the target container
  - dValue == a string of bytes that can be accommodated by the target container.

#### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. If the return code is BSI\_OK, the specified dValue is stored, with the specified tag, in the target container.
3. No other changes are made to the container structure of the connected card.

#### **Assertion 13.2**

Purpose: To test gscBsiGcDataCreate() using a bad handle.

#### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1302
  - the card has a target container
  - an authenticated session has been established with the target container.



2. A gscBsiGcDataCreate() call is made to the SPS, with
  - hCard != hCard1302
  - AID == the AID value of the target container
  - tag == a short whose value is not the tag of a data item in the target container
  - dValue == a string of bytes that can be accommodated by the target container.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

**Assertion 9.13.2.1**

**Assertion 9.13.2.2**

(See section 9.)

**Assertion 13.3**

Purpose: To test gscBsiGcDataCreate() using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1303
  - the card has a target container
  - an authenticated session has been established with the target container.
2. A gscBsiGcDataCreate() call is made to the SPS, with
  - hCard == hCard1303
  - AID == a String that does not contain the correct AID for any container on the connected card
  - tag == a short whose value is not the tag of a data item in the target container
  - dValue == a string of bytes that can be accommodated by the target container.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_AID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

**Assertion 9.13.3.1**

#### **Assertion 9.13.3.2**

(See section 9.)

#### **Assertion 13.4**

Purpose: To test gscBsiGcDataCreate() using a bad parameter.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1304
  - the card has a target container
  - an authenticated session has been established with the target container.
2. A gscBsiGcDataCreate() call is made to the SPS, with
  - hCard == hCard1304
  - AID == the AID value of the target container
  - tag == a short whose value is invalid for a tag
  - dValue == a string of bytes that can be accommodated by the target container.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_PARAM (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_PARAM) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

#### **Assertion 9.13.4.1**

#### **Assertion 9.13.4.2**

(See section 9.)

#### **Assertion 13.5**

Purpose: To test gscBsiGcDataCreate() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1305.
  - the card has a target container
  - an authenticated session has been established with the target container.
3. A gscBsiGcDataCreate() call is made to the SPS, with
  - hCard == hCard1305

- AID == the AID value of the target container
- tag == a short whose value is not the tag of a data item in the target container
- dValue == a string of bytes that can be accommodated by the target container.

Expected Results:

1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

**Assertion 9.13.5.1**

**Assertion 9.13.5.2**

(See section 9.)

**Assertion 13.6**

Purpose: To test gscBsiGcDataCreate() with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1306
  - the card has a target container
  - an authenticated session has been established with the target container.
2. The connected card is removed from the reader.
3. A gscBsiGcDataCreate() call is made to the SPS, with
  - hCard == hCard1306
  - AID == the AID value of the target container
  - tag == a short whose value is not the tag of a data item in the target container
  - dValue == a string of bytes that can be accommodated by the target container.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_REMOVED).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

**Assertion 13.7**

Purpose: To test gscBsiGcDataCreate() without fulfilling the applicable ACR.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1307
  - the card has a target container
  - an authenticated session has not been established with the target container.
2. A gscBsiGcDataCreate() call is made to the SPS, with
  - hCard == hCard1307
  - AID == the AID value of the target container
  - tag == a short whose value is not the tag of a data item in the target container
  - dValue == a string of bytes that can be accommodated by the target container.

Expected Results:

1. The call returns the return code BSI\_ACCESS\_DENIED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_ACCESS\_DENIED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

**Assertion 9.13.7.1**

**Assertion 9.13.7.2**

(See section 9.)

**Assertion 13.8**

Purpose: To test gscBsiGcDataCreate() using a too-large data value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1308
  - the card has a target container
  - an authenticated session has been established with the target container.
2. A gscBsiGcDataCreate() call is made to the SPS, with
  - hCard == hCard1308
  - AID == the AID value of the target container
  - tag == a short whose value is not the tag of a data item in the target container
  - dValue == a string of bytes that is too large to be accommodated by the target container.

Expected Results:

1. The call returns
  - the return code BSI\_NO\_MORE\_SPACE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_MORE\_SPACE) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

2. No changes are made to the container structure of the connected card.

**Assertion 9.13.8.1**

**Assertion 9.13.8.2**

(See section 9.)

**Assertion 13.9**

Purpose: To test gscBsiGcDataCreate() using the tag of a data item that already exists.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1309
  - the card has a target container
  - the target container has enough available space to accommodate dValue1309
  - an authenticated session has been established with the target container.
2. A gscBsiGcDataCreate() call is made to the SPS, with
  - hCard == hCard1309
  - AID == the AID value of the target container
  - tag == a short whose value is the tag of a data item in the target container
  - dValue == a string of bytes that can be accommodated by the target container.

Expected Results:

1. The call returns
  - the return code BSI\_TAG\_EXISTS (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_TAG\_EXISTS) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

**Assertion 9.13.9.1**

**Assertion 9.13.9.2**

(See section 9.)

## 14. gscBsiGcDataDelete()

```
public abstract void gscBsiGcDataDelete(  
    int          hCard,  
    String       AID,  
    short        tag  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.6.2.
2. GSC-IS F.3.2.

### **Assertion 14.1**

Purpose: To test gscBsiGcDataDelete() using valid parameters.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1401
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A gscBsiGcDataDelete() call is made to the SPS, with
  - hCard == hCard1401
  - AID == the AID value of the target container
  - tag == a short whose value is the tag of a data item in the target container.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. If the return code is BSI\_OK, then there is no longer a data item with the the specified tag stored in the target container.
3. No other changes are made to the container structure of the connected card.

### **Assertion 14.2**

Purpose: To test gscBsiGcDataDelete() using a bad handle.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1402
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A gscBsiGcDataDelete() call is made to the SPS, with

- hCard != hCard1402
- AID == the AID value of the target container
- tag == a short whose value is the tag of a data item in the target container.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

**Assertion 9.14.2.1**

**Assertion 9.14.2.2**

(See section 9.)

**Assertion 14.3**

Purpose: To test gscBsiGcDataDelete() using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1403
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A gscBsiGcDataDelete() call is made to the SPS, with
  - hCard == hCard1403
  - AID == a String that does not contain the correct AID for any container on the connected card
  - tag == a short whose value is the tag of a data item in the target container.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_AID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

**Assertion 9.14.3.1**

**Assertion 9.14.3.2**

(See section 9.)

#### **Assertion 14.4**

Purpose: To test `gscBsiGcDataDelete()` using a bad tag.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard1404`
  - the card has a target container
  - an authenticated session has been established with the target container.
2. A `gscBsiGcDataDelete()` call is made to the SPS, with
  - `hCard == hCard1404`
  - `AID ==` the AID value of the target container
  - `tag ==` a short whose value is not the tag of a data item in the target container.

Expected Results:

1. The call returns
  - the return code `BSI_BAD_TAG` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_BAD_TAG`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).
2. No changes are made to the container structure of the connected card.

#### **Assertion 9.14.4.1**

#### **Assertion 9.14.4.2**

(See section 9.)

#### **Assertion 14.5**

Purpose: To test `gscBsiGcDataDelete()` with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard1405`
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
3. A `gscBsiGcDataDelete()` call is made to the SPS, with
  - `hCard == hCard1405`
  - `AID ==` the AID value of the target container
  - `tag ==` a short whose value is the tag of a data item in the target container.

Expected Results:



1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

#### **Assertion 9.14.5.1**

#### **Assertion 9.14.5.2**

(See section 9.)

### **Assertion 14.6**

Purpose: To test gscBsiGcDataDelete() with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1406
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. The connected card is removed from the reader.
3. A gscBsiGcDataDelete() call is made to the SPS, with
  - hCard == hCard1406
  - AID == the AID value of the target container
  - tag == a short whose value is the tag of a data item in the target container.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_REMOVED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

### **Assertion 14.7**

Purpose: To test gscBsiGcDataDelete() without fulfilling the applicable ACR.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1407
  - the card has a target container, which contains at least one data item
  - an authenticated session has not been established with the target container.

2. A `gscBsiGcDataDelete()` call is made to the SPS, with
  - `hCard == hCard1407`
  - `AID ==` the AID value of the target container
  - `tag ==` a short whose value is the tag of a data item in the target container.

Expected Results:

1. The call returns
  - the return code `BSI_ACCESS_DENIED` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_ACCESS_DENIED`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).
2. No changes are made to the container structure of the connected card.

**Assertion 9.14.7.1**

**Assertion 9.14.7.2**

(See section 9.)

## 15. gscBsiGcGetContainerProperties()

```
public abstract ContainerProperties gscBsiGcGetContainerProperties(  
    int          hCard,  
    String       AID  
) throws gov.gsc.classes.BSIException;
```

The ContainerProperties class contains the fields

```
protected GCacr strctGCacr  
protected GCContainerSize strctContainerSizes  
protected String containerVersion
```

The GCacr class contains the fields

```
protected BSIacr createACR  
protected BSIacr deleteACR  
protected BSIacr readTagListACR  
protected BSIacr readValueACR  
protected BSIacr updateValueACR
```

The BSIacr class contains the fields

```
protected int ACRTYPE  
protected int[] keyIDOrReference  
protected int authNb  
protected int ACRID
```

The GCContainerSize class contains the fields

```
protected int maxNbDataItems  
protected int maxValueStorageSize
```

### References:

1. GSC-IS 4.6.3.
2. GSC-IS F.3.3.

### Starting state for each Assertion:

1. There exists a ContainerProperties object containerProps1500

### **Assertion 15.1**

Purpose: To test gscBsiGcGetContainerProperties() using valid parameters.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1501
  - the card has a target container.
2. A containerProps1500 = gscBsiGcGetContainerProperties() call is made to the SPS, with
  - hCard == hCard1501
  - AID == the AID value of the target container.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

2. The fields of the `strctGCacr` field of `containerProps1500` are correctly set to indicate access control conditions for all operations.
3. Within `containerProps1500`, for a Virtual Machine card, `maxNbDataItems` == the size of the T-Buffer, and `maxValueStorageSize` == the size of the V-Buffer. For file system cards that cannot calculate the size of the T- and V- buffers, `maxNbDataItems` == 0 and `maxValueStorageSize` == 0.
4. The `containerVersion` field of `containerProps1500` == the version of the target container.

## **Assertion 15.2**

Purpose: To test `gscBsiGcGetContainerProperties()` using a bad handle.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard1502`
  - the card has a target container.
2. A `containerProps1500 = gscBsiGcGetContainerProperties()` call is made to the SPS, with
  - `hCard != hCard1502`
  - `AID ==` the AID value of the target container.

Expected Results:

1. The call returns
  - the return code `BSI_BAD_HANDLE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_BAD_HANDLE`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

### **Assertion 9.15.2.1**

### **Assertion 9.15.2.2**

(See section 9.)

## **Assertion 15.3**

Purpose: To test `gscBsiGcGetContainerProperties()` using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard1503`
  - The card has a target container.
2. A `containerProps1500 = gscBsiGcGetContainerProperties()` call is made to the SPS, with
  - `hCard == hCard1503`
  - `AID ==` a String that does not contain the correct AID for any container on the connected card.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_AID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

#### **Assertion 9.15.3.1**

#### **Assertion 9.15.3.2**

(See section 9.)

### **Assertion 15.4**

Purpose: To test gscBsiGcGetContainerProperties() with another application having established a transaction lock.

#### Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1504
  - the card has a target container.
3. A containerProps1500 = gscBsiGcGetContainerProperties() call is made to the SPS, with
  - hCard == hCard1504
  - AID == the AID value of the target container.

#### Expected Results:

1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

### **Assertion 15.5**

Purpose: To test gscBsiGcGetContainerProperties() with a removed card.

#### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1505
  - The card has a target container.
2. The connected card is removed from the reader.
3. A containerProps1500 = gscBsiGcGetContainerProperties() call is made to the SPS, with
  - hCard == hCard1505
  - AID == the AID value of the target container.

#### Expected Results:

1. The call returns

- the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_REMOVED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

## 16. gscBsiGcReadTagList()

```
public abstract short[] gscBsiGcReadTagList(  
    int          hCard,  
    String       AID  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.6.4.
2. GSC-IS F.3.4.

### Starting state for each Assertion:

1. tagListArray1600 is an array of shorts.

### **Assertion 16.1**

Purpose: To test gscBsiGcReadTagList() using valid parameters.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1601
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A tagListArray1600 = gscBsiGcReadTagList() call is made to the SPS, with
  - hCard == hCard1601
  - AID == the AID value of the target container.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. If the return code is BSI\_OK, then tagListArray1600 == a Vector containing the list of tags for the target container.

### **Assertion 16.2**

Purpose: To test gscBsiGcReadTagList() using a bad handle.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1602
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A tagListArray1600 = gscBsiGcReadTagList() call is made to the SPS, with
  - hCard /= hCard1602

- AID == the AID value of the target container.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.16.2.1**

**Assertion 9.16.2.2**

(See section 9.)

**Assertion 16.3**

Purpose: To test gscBsiGcReadTagList() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1603
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
3. A tagListArray1600 = gscBsiGcReadTagList() call is made to the SPS, with
  - hCard == hCard1603
  - AID == the AID value of the target container.

Expected Results:

1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.16.3.1**

**Assertion 9.16.3.2**

(See section 9.)

**Assertion 16.4**

Purpose: To test gscBsiGcReadTagList() using a bad AID value.

Scenario:



1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1604
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A tagListArray1600 = gscBsiGcReadTagList() call is made to the SPS, with
  - hCard == hCard1604
  - AID == a String that does not contain the correct AID for any container on the connected card.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_AID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.16.4.1**

**Assertion 9.16.4.2**

(See section 9.)

**Assertion 16.5**

Purpose: To test gscBsiGcReadTagList() with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1605
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. The connected card is removed from the reader.
3. A tagListArray1600 = gscBsiGcReadTagList() call is made to the SPS, with
  - hCard == hCard1605
  - AID == the AID value of the target container.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_REMOVED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

## **Assertion 16.6**

Purpose: To test gscBsiGcReadTagList() without fulfilling the applicable ACR.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1606
  - the card has a target container, which contains at least one data item
  - an authenticated session has not been established with the target container.
2. A tagListArray1600 = gscBsiGcReadTagList() call is made to the SPS, with
  - hCard == hCard1606
  - AID == the AID value of the target container.

### Expected Results:

1. The call returns
  - the return code BSI\_ACCESS\_DENIED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_ACCESS\_DENIED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

#### **Assertion 9.16.6.1**

#### **Assertion 9.16.6.2**

(See section 9.)

## 17. gscBsiGcReadValue()

```
public abstract byte[] gscBsiGcReadValue(  
    int          hCard,  
    String       AID,  
    short        tag  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.6.5.
2. GSC-IS F.3.5.

### Starting state for each Assertion:

1. dValue1700 is an array of bytes.

### **Assertion 17.1**

Purpose: To test gscBsiGcReadValue() using valid parameters.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1701
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A dValue1700 = gscBsiGcReadValue() call is made to the SPS, with
  - hCard == hCard1701
  - AID == the AID value of the target container
  - tag == a short whose value is the tag of a data item in the target container.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. If the return code is BSI\_OK, then dValue1700 == an array of bytes containing the value associated with the specified tag.

### **Assertion 17.2**

Purpose: To test gscBsiGcReadValue() using a bad handle.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1702
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A dValue1700 = gscBsiGcReadValue() call is made to the SPS, with

- hCard != hCard1702
- AID == the AID value of the target container
- tag == a short whose value is the tag of a data item in the target container.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.17.2.1**

**Assertion 9.17.2.2**

(See section 9.)

**Assertion 17.3**

Purpose: To test gscBsiGcReadValue() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1703
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
3. A dValue1700 = gscBsiGcReadValue() call is made to the SPS, with
  - hCard == hCard1703
  - AID == the AID value of the target container
  - tag == a short whose value is the tag of a data item in the target container.

Expected Results:

1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.17.3.1**

**Assertion 9.17.3.2**

(See section 9.)

**Assertion 17.4**

Purpose: To test gscBsiGcReadValue() using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1704
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A dValue1700 = gscBsiGcReadValue() call is made to the SPS, with
  - hCard == hCard1704
  - AID == a String that does not contain the correct AID for any container on the connected card
  - tag == a short whose value is the tag of a data item in the target container.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_AID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.17.4.1**

**Assertion 9.17.4.2**

(See section 9.)

**Assertion 17.5**

Purpose: To test gscBsiGcReadValue() using a bad tag.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1705
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A dValue1700 = gscBsiGcReadValue() call is made to the SPS, with
  - hCard == hCard1705
  - AID == the AID value of the target container
  - tag == a short whose value is not the tag of a data item in the target container.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_TAG (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_TAG) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

#### **Assertion 9.17.5.1**

#### **Assertion 9.17.5.2**

(See section 9.)

### **Assertion 17.6**

Purpose: To test gscBsiGcReadValue() with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1706
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. The connected card is removed from the reader.
3. A dValue1700 = gscBsiGcReadValue() call is made to the SPS, with
  - hCard == hCard1706
  - AID == the AID value of the target container
  - tag == a short whose value is the tag of a data item in the target container.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_REMOVED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

### **Assertion 17.7**

Purpose: To test gscBsiGcReadValue() without fulfilling the applicable ACR.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1707
  - the card has a target container which contains at least one data item
  - an authenticated session has not been established with the target container.
2. A dValue1700 = gscBsiGcReadValue() call is made to the SPS, with
  - hCard == hCard1707
  - AID == the AID value of the target container

- tag == a short whose value is the tag of a data item in the target container.

Expected Results:

1. The call returns
  - the return code BSI\_ACCESS\_DENIED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_ACCESS\_DENIED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.17.7.1**

**Assertion 9.17.7.2**

(See section 9.)

## 18. gscBsiGcUpdateValue()

```
public abstract void gscBsiGcUpdateValue(  
    int          hCard,  
    String       AID,  
    short        tag,  
    byte[]       dValue  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.6.5.
2. GSC-IS F.3.6.

### Starting state for each Assertion:

1. dValue1800 is an array of bytes.

### **Assertion 18.1**

Purpose: To test gscBsiGcUpdateValue() using valid parameters.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1801
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A gscBsiGcUpdateValue() call is made to the SPS, with
  - hCard == hCard1801
  - AID == the AID value of the target container
  - tag == a short whose value is the tag of an existing data item in the target container
  - dValue == a new data item which is /= the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. If the return code is BSI\_OK, then the new data item is stored, with the specified tag, in the target container.
3. No other changes are made to the container structure of the connected card.

### **Assertion 18.2**

Purpose: To test gscBsiGcUpdateValue() using a bad handle.



Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1802
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A gscBsiGcUpdateValue() call is made to the SPS, with
  - hCard != hCard1802
  - AID == the AID value of the target container
  - tag == a short whose value is the tag of an existing data item in the target container
  - dValue == a new data item which is != the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

**Assertion 9.18.2.1**

**Assertion 9.18.2.2**

(See section 9.)

**Assertion 18.3**

Purpose: To test gscBsiGcUpdateValue() using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1803
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A gscBsiGcUpdateValue() call is made to the SPS, with
  - hCard == hCard1803
  - AID == a String that does not contain the correct AID for any container on the connected card
  - tag == a short whose value is the tag of an existing data item in the target container
  - dValue == a new data item which is != the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_AID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

**Assertion 9.18.3.1**

**Assertion 9.18.3.2**

(See section 9.)

**Assertion 18.4**

Purpose: To test gscBsiGcUpdateValue() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1804
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
3. A gscBsiGcUpdateValue() call is made to the SPS, with
  - hCard == hCard1804
  - AID == the AID value of the target container
  - tag == a short whose value is the tag of an existing data item in the target container
  - dValue == a new data item which is /= the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item.

Expected Results:

1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.18.4.1**

**Assertion 9.18.4.2**

(See section 9.)

**Assertion 18.5**

Purpose: To test gscBsiGcUpdateValue() using a bad parameter.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1805
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container
4. A gscBsiGcUpdateValue() call is made to the SPS, with
  - hCard == hCard1805
  - AID == the AID value of the target container
  - tag == a short whose value is invalid for a tag
  - dValue == an arbitrary data item.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_PARAM (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_PARAM) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

**Assertion 9.18.5.1**

**Assertion 9.18.5.2**

(See section 9.)

**Assertion 18.6**

Purpose: To test gscBsiGcUpdateValue() using a bad tag.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1806
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A gscBsiGcUpdateValue() call is made to the SPS, with
  - hCard == hCard1806
  - AID == the AID value of the target container
  - tag == a short whose value is not the tag of an existing data item in the target container
  - dValue == an arbitrary data item.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_TAG (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_TAG) or the return code

BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

2. No changes are made to the container structure of the connected card.

#### **Assertion 9.18.6.1**

#### **Assertion 9.18.6.2**

(See section 9.)

### **Assertion 18.7**

Purpose: To test gscBsiGcUpdateValue() with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1807
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. The connected card is removed from the reader.
3. A gscBsiGcUpdateValue() call is made to the SPS, with
  - hCard == hCard1807
  - AID == the AID value of the target container
  - tag == a short whose value is the tag of an existing data item in the target container
  - dValue == a new data item which is /= the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_REMOVED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

### **Assertion 18.8**

Purpose: To test gscBsiGcUpdateValue() without fulfilling the applicable ACR.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1808
  - the card has a target container, which contains at least one data item

- an authenticated session has not been established with the target container.
2. A gscBsiGcUpdateValue() call is made to the SPS, with
    - hCard == hCard1808
    - AID == the AID value of the target container
    - tag == a short whose value is the tag of an existing data item in the target container
    - dValue == a new data item which is /= the existing data item identified by the specified tag in the target container, but which could be accommodated by the container if it replaced the existing data item.

Expected Results:

1. The call returns
  - the return code BSI\_ACCESS\_DENIED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_ACCESS\_DENIED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. No changes are made to the container structure of the connected card.

**Assertion 9.18.8.1**

**Assertion 9.18.8.2**

(See section 9.)

**Assertion 18.9**

Purpose: To test gscBsiGcUpdateValue() using a too-large data value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1809
  - the card has a target container, which contains at least one data item
  - an authenticated session has been established with the target container.
2. A gscBsiGcUpdateValue() call is made to the SPS, with
  - hCard == hCard1809
  - AID == the AID value of the target container
  - tag == a short whose value is the tag of a data item in the target container
  - dValue == a new data item which is /= the existing data item identified by the specified tag, and which could not be accommodated by the target container if it were to replace the existing data item.

Expected Results:

1. The call returns
  - the return code BSI\_NO\_MORE\_SPACE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_MORE\_SPACE) or the

return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

2. No changes are made to the container structure of the connected card.

**Assertion 9.18.9.1**

**Assertion 9.18.9.2**

(See section 9.)

## 19. gscBsiGetChallenge()

```
public abstract byte[] gscBsiGetChallenge(  
    int          hCard,  
    String       AID  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.7.1.
2. GSC-IS F.4.1.

### Starting state for each Assertion:

1. challenge1900 is an array of bytes.

### **Assertion 19.1**

Purpose: To test gscBsiGetChallenge() using valid parameters

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1901
  - the card has a target container.
2. A challenge1900 = gscBsiGetChallenge() call is made to the SPS, with
  - hCard == hCard1901
  - AID == the AID value of the target container.

### Expected Results:

1. The call returns:
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. If the return code is BSI\_OK, then challenge1900 == an array of bytes containing the random challenge returned from the connected card.

### **Assertion 19.2**

Purpose: To test gscBsiGetChallenge() using a bad handle.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1902
  - the card has a target container.
2. A challenge1900 = gscBsiGetChallenge() call is made to the SPS, with
  - hCard != hCard1902
  - AID == the AID value of the target container.

### Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

#### **Assertion 9.19.2.1**

#### **Assertion 9.19.2.2**

(See section 9.)

### **Assertion 19.3**

Purpose: To test gscBsiGetChallenge() using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1903
  - the card has a target container.
2. A challenge1900 = gscBsiGetChallenge() call is made to the SPS, with
  - hCard == hCard1903
  - AID == a String that does not contain the correct AID for any container on the connected card.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_AID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

#### **Assertion 9.19.3.1**

#### **Assertion 9.19.3.2**

(See section 9.)

### **Assertion 19.4**

Purpose: To test gscBsiGetChallenge() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1904
  - the card has a target container.
3. A challenge1900 = gscBsiGetChallenge() call is made to the SPS, with
  - hCard == hCard1904
  - AID == the AID value of the target container.

Expected Results:

1. The call returns



- the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

#### **Assertion 9.19.4.1**

#### **Assertion 9.19.4.2**

(See section 9.)

### **Assertion 19.5**

Purpose: To test gscBsiGetChallenge() with a removed card.

#### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1905
  - the card has a target container.
2. The connected card is removed from the reader.
3. A challenge1900 = gscBsiGetChallenge() call is made to the SPS, with
  - hCard == hCard1905
  - AID == the AID value of the target container.

#### Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_REMOVED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

## 20. gscBsiSkiInternalAuthenticate()

```
public abstract byte[] gscBsiSkiInternalAuthenticate(  
    int          hCard,  
    String       AID,  
    short        algoID,  
    byte[]       challenge  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.7.2
2. GSC-IS F.4.2.

### Starting state for each Assertion:

1. cryptogram2000 is an array of bytes.

### **Assertion 20.1**

Purpose: To test gscBsiSkiInternalAuthenticate() using valid parameters.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2001
  - the card has an target SKI provider module
  - an authenticated session has been established with the target container.
2. A cryptogram2000 = gscBsiSkiInternalAuthenticate() call is made to the SPS, with
  - hCard == hCard2001
  - AID == the AID value of the target SKI provider module
  - algoID == a short that represents a valid cryptographic algorithm that the connected card must use to encrypt the challenge
  - challenge == an array of bytes containing the challenge submitted to the connected card.

### Expected Results:

1. The call returns:
  - the return code BSI\_OK (no BSIException is thrown), BSI\_TERMINAL\_AUTH (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_TERMINAL\_AUTH) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. If the return code is BSI\_OK, then cryptogram2000 == the cryptogram computed by the connected card.

### **Assertion 20.2**

Purpose: To test gscBsiSkiInternalAuthenticate() using a bad handle.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2002

- the card has an target SKI provider module
  - an authenticated session has been established with the target container.
2. A `cryptogram2000 = gscBsiSkiInternalAuthenticate()` call is made to the SPS, with
- `hCard != hCard2002`
  - `AID ==` the AID value of the target SKI provider module
  - `algoID ==` a short that represents a valid cryptographic algorithm that the connected card must use to encrypt the challenge
  - `challenge ==` an array of bytes containing the challenge submitted to the connected card.

Expected Results:

1. The call returns
  - the return code `BSI_BAD_HANDLE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_BAD_HANDLE`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

**Assertion 9.20.2.1**

**Assertion 9.20.2.2**

(See section 9.)

**Assertion 20.3**

Purpose: To test `gscBsiSkiInternalAuthenticate()` using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard2003`
  - the card has an target SKI provider module
  - an authenticated session has been established with the target container.
2. A `cryptogram2000 = gscBsiSkiInternalAuthenticate()` call is made to the SPS, with
  - `hCard == hCard2003`
  - `AID ==` a String that does not contain the correct AID for any container on the connected card
  - `algoID ==` a short that represents a valid cryptographic algorithm that the connected card must use to encrypt the challenge
  - `challenge ==` an array of bytes containing the challenge submitted to the connected card.

Expected Results:

1. The call returns
  - the return code `BSI_BAD_AID` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_BAD_AID`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

#### **Assertion 9.20.3.1**

#### **Assertion 9.20.3.2**

(See section 9.)

### **Assertion 20.4**

Purpose: To test `gscBsiSkiInternalAuthenticate()` with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard2004`
  - the card has an target SKI provider module
  - an authenticated session has been established with the target container.
3. A `cryptogram2000 = gscBsiSkiInternalAuthenticate()` call is made to the SPS, with
  - `hCard == hCard2004`
  - `AID ==` the AID value of the target SKI provider module
  - `algoID ==` a short that represents a valid cryptographic algorithm that the connected card must use to encrypt the challenge
  - `challenge ==` an array of bytes containing the challenge submitted to the connected card.

Expected Results:

1. The call returns
  - the return code `BSI_SC_LOCKED` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_SC_LOCKED`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

#### **Assertion 9.20.4.1**

#### **Assertion 9.20.4.2**

(See section 9.)

### **Assertion 20.5**

Purpose: To test `gscBsiSkiInternalAuthenticate()` using a bad parameter.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard2005`
  - the card has an target SKI provider module
  - an authenticated session has been established with the target container.

2. A `cryptogram2000 = gscBsiSkiInternalAuthenticate()` call is made to the SPS, with
  - `hCard == hCard2005`
  - `AID ==` the AID value of the target SKI provider module
  - `algoID ==` a short that represents a valid cryptographic algorithm that the connected card must use to encrypt the challenge
  - `challenge ==` an array of bytes that does not represent a possible challenge to the connected card.

Expected Results:

1. The call returns
  - the return code `BSI_BAD_PARAM` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_BAD_PARAM`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

**Assertion 9.20.5.1**

**Assertion 9.20.5.2**

(See section 9.)

**Assertion 20.6**

Purpose: To test `gscBsiSkiInternalAuthenticate()` using a bad cryptographic algorithm identifier.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard2006`
  - the card has an target SKI provider module
  - an authenticated session has been established with the target container.
2. A `cryptogram2000 = gscBsiSkiInternalAuthenticate()` call is made to the SPS, with
  - `hCard == hCard2006`
  - `AID ==` the AID value of the target SKI provider module
  - `algoID ==` a short that does not represent a valid cryptographic algorithm for the connected card to use to encrypt the challenge
  - `challenge ==` an array of bytes containing the challenge submitted to the connected card.

Expected Results:

1. The call returns
  - the return code `BSI_BAD_ALGO_ID` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_BAD_ALGO_ID`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

**Assertion 9.20.6.1**

**Assertion 9.20.6.2**

(See section 9.)

#### **Assertion 20.7**

Purpose: To test `gscBsiSkiInternalAuthenticate()` with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard2007`
  - the card has an target SKI provider module
  - an authenticated session has been established with the target container.
2. The connected card is removed from the reader.
3. A `cryptogram2000 = gscBsiSkiInternalAuthenticate()` call is made to the SPS, with
  - `hCard == hCard2007`
  - `AID ==` the AID value of the target SKI provider module
  - `algoID ==` a short that represents a valid cryptographic algorithm that the connected card must use to encrypt the challenge
  - `challenge ==` an array of bytes containing the challenge submitted to the connected card.

Expected Results:

1. The call returns
  - the return code `BSI_CARD_REMOVED` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_CARD_REMOVED`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of `GetExtendedErrorText()`.*

#### **Assertion 20.8**

Purpose: To test `gscBsiSkiInternalAuthenticate()` without fulfilling the applicable ACR.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard2008`
  - the card has a target SKI provider module
  - an authenticated session has not been established with the target container.
2. A `cryptogram2000 = gscBsiSkiInternalAuthenticate()` call is made to the SPS, with
  - `hCard == hCard2008`
  - `AID ==` the AID value of the target container
  - `algoID ==` a short that represents a valid cryptographic algorithm that the connected card must use to encrypt the challenge
  - `challenge ==` an array of bytes containing the challenge submitted to the connected card.

Expected Results:

1. The call returns
  - the return code BSI\_ACCESS\_DENIED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_ACCESS\_DENIED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.20.8.1**

**Assertion 9.20.8.2**

(See section 9.)

## 21. gscBsiPkiCompute()

```
public abstract byte[] gscBsiPkiCompute(  
    int          hCard,  
    String       AID,  
    short        algoID,  
    byte[]       message  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.7.3.
2. GSC-IS F.4.3.

### Starting state for each Assertion:

1. result2100 is an array of bytes.

### **Assertion 21.1**

Purpose: To test gscBsiPkiCompute() using valid parameters.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2101
  - the card has a target PKI provider module
  - an authenticated session has been established with the target container.
2. A result2100 = gscBsiPkiCompute() call is made to the SPS, with
  - hCard == hCard2101
  - AID == the AID value of the target PKI provider module
  - algoID == a short that represents the cryptographic algorithm that will be used to generate the signature
  - message == an array of bytes containing the hash of the message to be signed.

### Expected Results:

1. The call returns:
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. If the return code is BSI\_OK, then result2100 == an array of bytes containing the signature generated by the connected card.

### **Assertion 21.2**

Purpose: To test gscBsiPkiCompute() using a bad handle.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2102
  - the card has a target PKI provider module
  - an authenticated session has been established with the target container.



2. A result2100 = gscBsiPkiCompute() call is made to the SPS, with
  - hCard != hCard2102
  - AID == the AID value of the target PKI provider module
  - algoID == a short that represents the cryptographic algorithm that will be used to generate the signature
  - message == an array of bytes containing the hash of the message to be signed.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.21.2.1**

**Assertion 9.21.2.2**

(See section 9.)

**Assertion 21.3**

Purpose: To test gscBsiPkiCompute() using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2103
  - the card has a target PKI provider module
  - an authenticated session has been established with the target container.
2. A result2100 = gscBsiPkiCompute() call is made to the SPS, with
  - hCard == hCard2103
  - AID == a String that does not contain the correct AID for any container on the connected card
  - algoID == a short that represents the cryptographic algorithm that will be used to generate the signature
  - message == an array of bytes containing the hash of the message to be signed.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_AID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.21.3.1**

**Assertion 9.21.3.2**

(See section 9.)

#### **Assertion 21.4**

Purpose: To test `gscBsiPkiCompute()` with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard2104`
  - the card has a target PKI provider module
  - an authenticated session has been established with the target container.
3. A `result2100 = gscBsiPkiCompute()` call is made to the SPS, with
  - `hCard == hCard2104`
  - `AID ==` the AID value of the target PKI provider module
  - `algoID ==` a short that represents the cryptographic algorithm that will be used to generate the signature
  - `message ==` an array of bytes containing the hash of the message to be signed.

Expected Results:

1. The call returns
  - the return code `BSI_SC_LOCKED` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_SC_LOCKED`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

#### **Assertion 9.21.4.1**

#### **Assertion 9.21.4.2**

(See section 9.)

#### **Assertion 21.5**

Purpose: To test `gscBsiPkiCompute()` using a bad parameter.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle `hCard2105`
  - the card has a target PKI provider module
  - an authenticated session has been established with the target container.
2. A `result2100 = gscBsiPkiCompute()` call is made to the SPS, with
  - `hCard == hCard2105`
  - `AID ==` the AID value of the target PKI provider module
  - `algoID ==` a short that represents the cryptographic algorithm that will be used to generate the signature
  - `message ==` an array of bytes that does not represent a possible hash of the message to be signed.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_PARAM (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_PARAM) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.21.5.1**

**Assertion 9.21.5.2**

(See section 9.)

**Assertion 21.6**

Purpose: To test gscBsiPkiCompute() using a bad cryptographic algorithm identifier.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2106
  - the card has a target PKI provider module
  - an authenticated session has been established with the target container.
2. A result2100 = gscBsiPkiCompute() call is made to the SPS, with
  - hCard == hCard2106
  - AID == the AID value of the target PKI provider module
  - algoID == an array of bytes that contains an invalid cryptographic algorithm for the connected card to use to generate the signature.
  - message == an array of bytes containing the hash of the message to be signed.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_ALGO\_ID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_ALGO\_ID) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.21.6.1**

**Assertion 9.21.6.2**

(See section 9.)

**Assertion 21.7**

Purpose: To test gscBsiPkiCompute() with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2107

- the card has a target PKI provider module
  - an authenticated session has been established with the target container.
2. The connected card is removed from the reader.
  3. A result2100 = gscBsiPkiCompute() call is made to the SPS, with
    - hCard == hCard2107
    - AID == the AID value of the target PKI provider module
    - algoID == a short that represents the cryptographic algorithm that will be used to generate the signature
    - message == an array of bytes containing the hash of the message to be signed.

Expected Results:

1. The call returns
  - the return code BSI\_CARD\_REMOVED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_CARD\_REMOVED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().*

**Assertion 21.8**

Purpose: To test gscBsiPkiCompute() without fulfilling the applicable ACR.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2108
  - the card has a target PKI provider module
  - an authenticated session has not been established with the target container.
2. A result2100 = gscBsiPkiCompute() call is made to the SPS, with
  - hCard == hCard2108
  - AID == the AID value of the target container
  - algoID == a short that represents the cryptographic algorithm that will be used to generate the signature
  - message == an array of bytes containing the hash of the message to be signed.

Expected Results:

1. The call returns
  - the return code BSI\_ACCESS\_DENIED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_ACCESS\_DENIED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.21.8.1**

**Assertion 9.21.8.2**

(See section 9.)

## 22. gscBsiPkiGetCertificate()

```
public abstract byte[] gscBsiPkiGetCertificate(  
    int          hCard,  
    String       AID  
) throws gov.gsc.classes.BSIException;
```

### References:

1. GSC-IS 4.7.4.
2. GSC-IS F.4.4.

### Starting state for each Assertion:

1. certificate2200 is an array of bytes.

### **Assertion 22.1**

Purpose: To test gscBsiPkiGetCertificate() using valid parameters.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with the handle hCard2201
  - the card has a target PKI provider module
  - an authenticated session has been established with the target container.
2. A certificate2200 = gscBsiPkiGetCertificate() call is made to the SPS, with
  - hCard == hCard2201
  - AID == the AID value of the target PKI provider module.

### Expected Results:

1. The call returns:
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. If the return code is BSI\_OK, then certificate2200 == an array of bytes containing the certificate returned from the connected card.

### **Assertion 22.2**

Purpose: To test gscBsiPkiGetCertificate() using a bad handle.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with the handle hCard2202
  - the card has a target container
  - an authenticated session has been established with the target container.
2. A certificate2200 = gscBsiPkiGetCertificate() call is made to the SPS, with
  - hCard /= hCard2202
  - AID == the AID value of the target PKI provider module.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.22.2.1**

**Assertion 9.22.2.2**

(See section 9.)

**Assertion 22.3**

Purpose: To test gscBsiPkiGetCertificate() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with the handle hCard2203
  - the card has a target container
  - an authenticated session has been established with the target container.
3. A certificate2200 = gscBsiPkiGetCertificate() call is made to the SPS, with
  - hCard == hCard2203
  - AID == the AID value of the target PKI provider module.

Expected Results:

1. The call returns
  - the return code BSI\_SC\_LOCKED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_SC\_LOCKED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.22.3.1**

**Assertion 9.22.3.2**

(See section 9.)

**Assertion 22.4**

Purpose: To test gscBsiPkiGetCertificate() using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with the handle hCard2204
  - the card has a target container

- an authenticated session has been established with the target container.
2. A `certificate2200 = gscBsiPkiGetCertificate()` call is made to the SPS, with
    - `hCard == hCard2204`
    - `AID ==` a String that does not contain the correct AID for any container on the connected card.

Expected Results:

1. The call returns
  - the return code `BSI_BAD_AID` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_BAD_AID`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

**Assertion 9.22.4.1**

**Assertion 9.22.4.2**

(See section 9.)

**Assertion 22.5**

Purpose: To test `gscBsiPkiGetCertificate()` with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with the handle `hCard2205`
  - the card has a target container
  - an authenticated session has been established with the target container.
2. The connected card is removed from the reader.
3. A `certificate2200 = gscBsiPkiGetCertificate()` call is made to the SPS, with
  - `hCard == hCard2205`
  - `AID ==` the AID value of the target PKI provider module.

Expected Results:

1. The call returns
  - the return code `BSI_CARD_REMOVED` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_CARD_REMOVED`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of `GetExtendedErrorText()`.*

**Assertion 22.6**



Purpose: To test gscBsiPkiGetCertificate() without fulfilling the applicable ACR.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with the handle hCard2206
  - the card has a target container
  - an authenticated session has not been established with the target container.
2. A certificate2200 = gscBsiPkiGetCertificate() call is made to the SPS, with
  - hCard == hCard2206
  - AID == the AID value of the target container.

Expected Results:

1. The call returns
  - the return code BSI\_ACCESS\_DENIED (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_ACCESS\_DENIED) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.22.6.1**

**Assertion 9.22.6.2**

(See section 9.)

## 23. gscBsiGetCryptoProperties()

```
public abstract CryptoProperties gscBsiGetCryptoProperties(  
    int          hCard,  
    String       AID  
) throws gov.gsc.classes.BSIException;
```

The CryptoProperties class contains the fields  
protected CRYPTOacr strctCRYPTOacr  
protected int keyLen

The CRYPTOacr class contains the fields  
protected BSIacr getChallengeACR  
protected BSIacr internalAuthenticateACR  
protected BSIacr pkiComputeAcr  
  
protected BSIacr createACR  
protected BSIacr deleteACR  
protected BSIacr readTagListACR  
protected BSIacr readValueACR  
protected BSIacr updateValueAcr

### References:

1. GSC-IS 4.7.5.
2. GSC-IS F.4.5.

### Starting state for each Assertion:

1. CryptoProperties2300 is a CryptoProperties object.

## Assertion 23.1

Purpose: To test gscBsiGetCryptoProperties() using valid parameters.

### Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with the handle hCard2301
  - the card has a target PKI provider module.
2. A CryptoProperties2300 = gscBsiGetCryptoProperties() call is made to the SPS, with
  - hCard == hCard2301
  - AID == the AID value of the target PKI provider module.

### Expected Results:

1. The call returns
  - the return code BSI\_OK (no BSIException is thrown) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).
2. If the return code is BSI\_OK, then the fields of CryptoProperties2300 are correctly set to indicate access control conditions for all operations.

## Assertion 23.2

Purpose: To test gscBsiGetCryptoProperties() using a bad handle.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with the handle hCard2302
  - the card has a target PKI provider module.
2. A CryptoProperties = gscBsiGetCryptoProperties() call is made to the SPS, with
  - hCard /= hCard2302
  - AID == the AID value of the target PKI provider module.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_HANDLE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_HANDLE) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.23.2.1**

**Assertion 9.23.2.2**

(See section 9.)

**Assertion 23.3**

Purpose: To test gscBsiGetCryptoProperties() using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with the handle hCard2303
  - the card has a target PKI provider module.
2. A gscBsiGetCryptoProperties() call is made to the SPS, with
  - hCard == hCard2303
  - AID == a String that does not contain the correct AID for any container on the connected card.

Expected Results:

1. The call returns
  - the return code BSI\_BAD\_AID (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_BAD\_AID) or the return code BSI\_NO\_CARDSERVICE (a BSIException is thrown, with BSIException.getErrorCode returning BSI\_NO\_CARDSERVICE).

**Assertion 9.23.3.1**

**Assertion 9.23.3.2**

(See section 9.)

**Assertion 23.4**

Purpose: To test `gscBsiGetCryptoProperties()` with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with the handle `hCard2304`
  - the card has a target PKI provider module.
3. A `CryptoProperties = gscBsiGetCryptoProperties()` call is made to the SPS, with
  - `hCard == hCard2304`
  - `AID ==` the AID value of the target PKI provider module.

Expected Results:

1. The call returns
  - the return code `BSI_SC_LOCKED` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_SC_LOCKED`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

**Assertion 9.23.4.1**

**Assertion 9.23.4.2**

(See section 9.)

**Assertion 23.5**

Purpose: To test `gscBsiGetCryptoProperties()` with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with the handle `hCard2305`
  - the card has a target PKI provider module.
2. The connected card is removed from the reader.
3. A `gscBsiGetCryptoProperties()` call is made to the SPS, with
  - `hCard == hCard2305`
  - `AID ==` the AID value of the target PKI provider module.

Expected Results:

1. The call returns
  - the return code `BSI_CARD_REMOVED` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_CARD_REMOVED`) or the return code `BSI_NO_CARDSERVICE` (a `BSIException` is thrown, with `BSIException.getErrorCode` returning `BSI_NO_CARDSERVICE`).

*Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of `GetExtendedErrorText()`.*

**Substantive changes from the January 22 version of this document**

1. 20.5 and 21.5 now generate the BAD\_PARAM by using a bad challenge or message parameter, respectively.